TRANSCODING" and "ELECTRONIC DOCUMENT DELIVERY SYSTEM EMPLOYING DISTRIBUTED DOCUMENT OBJECT MODEL (DOM) BASED TRANSCODING AND PROVIDING ASSISTIVE TECHNOLOGY SUPPORT" by Richard S. Schwerdtfeger, Lawrence F. Weiss, and Rabindranath Dutta.

Please replace pg. 6, line 27 - pg. 7, line 4, with the amended paragraph below. A "marked-up" version of each amendment is included in **Attachment A**.

The transcoder receives the first portion of the electronic document in the first digital format, and translates the first portion of the electronic document from the first digital format to the original script in the second digital format. The original script includes the element and the identifier assigned to the element. As described above, the original script may also include the code value associated with the JAVASCRIPT event. The code value may be associated with the element within the original script. The transcoder provides the original script (e.g., to a client machine).

Please replace pg. 11, line 25 - pg. 12, line 7, with the amended paragraph below. A "marked-up" version of each amendment is included in **Attachment A**.

In response to electronic document 12 provided by internet server 16, synchronous DOM generator produces a pre-transcoded DOM 38. Electronic document 12 includes one or more elements representing document structures. Examples of document elements include paragraphs, hypertext links, lists, tables, and images. ID generator 36 assigns a unique identifier to each element of electronic document 12. Pre-transcoded DOM 38 defines a tree-like logical structure of document 12, and also defines methods for accessing and manipulating document 12. Within pre-transcoded DOM 38, elements of electronic document 12 are associated with their identifiers. As will be described in more detail below, pre-transcoded DOM 38 facilitates navigation of the logical structure of electronic document 12. Pre-transcoded DOM 38 preferably complies with the World Wide Web Consortium (W3C) DOM specification available on the Web at URL http://www.w3.org/.

Please replace pg. 12, lines 16-23, with the amended paragraph below. A "marked-up" version of each amendment is included in **Attachment A**.

The scripting language may be, for example, a subset of the digital format in which transcoder proxy 28 receives document 12 from internet server 16 (e.g., a subset of HTML, XML, POSTSCRIPT, PDF, or AFP). In producing the script, transcoder proxy 28 may convert graphics images within electronic document 12 from one format to another (e.g., from joint photographic experts group/JPEG format to graphics interchange format/GIF format, from JPEG and GIF formats to scaled vector graphics/SVG format, etc.). The script may also include audio data files (e.g., wav files), transcoded Braille, and/or unformatted text.

Please replace pg. 17, line 27 - pg. 18, line 8, with the amended paragraph below. A "marked-up" version of each amendment is included in **Attachment A.**

On the other hand, JAVASCRIPT engine 70 may recognize the mouseover event information and respond to the mouseover event by accessing the element associated with the mouseover event within pre-transcoded DOM 38 using the identifier. JAVASCRIPT engine 70 may use the code value to access the JAVASCRIPT code of the element. JAVASCRIPT engine 70 executes the JAVASCRIPT code of the element, thus producing the onmouseover data. JAVASCRIPT engine 70 may provide the onmouseover data to synchronous DOM generator 30. Synchronous DOM generator 30 may use pre-transcoded DOM 38 and the onmouseover data to produce a modified portion (e.g., a second portion) of document 12. Synchronous DOM generator 30 may provide the second portion of document 12, including the "onmouseover" data to transcoder 32.

Please replace pg. 21, lines 8-18, with the amended paragraph below. A "marked-up" version of each amendment is included in **Attachment A.**

Assistive technology 74 within client machine 22 may cooperate with assistive technology interface generator 72 within transcoder proxy 28. For example, where assistive technology 74 may be a screen reader, and the user may press a button in order to make the screen reader "talk faster". In response, assistive technology 74 may produce "talk faster" request signals, and may provide the "talk faster" request signals to user agent 40. User agent 40 may forward the "talk faster" request signals to assistive technology interface generator 72. Assistive technology interface generator 72 may respond to the "talk faster" request signals by producing one or more commands which, when executed by assistive technology 74, cause the screen reader to produce speech at a faster rate. Assistive technology interface generator 72 may provide the commands to assistive technology 74 via user agent 40.